Some selected formulas in the course SF2862, March 2009

This formula-sheet will be available at the exam in SF2862. Don't bring it yourself!

Important note: **No calculator on the exam!**

Some quantities and relations in queueing theory:
$L = \sum_{n=0}^{\infty} nP_n\,,\ L_q = \sum_{n=s}^{\infty}(n-s)P_n\,,\ \bar\lambda = \sum_{n=0}^{\infty}\lambda_n P_n,\ L = \bar\lambda W\,,\ L_q = \bar\lambda W_q\,.$

Balance equations for the birth-and-death process:
$\mu_{n+1}P_{n+1} = \lambda_n P_n,$ for $n = 0, 1, \ldots,$ and $\sum_{n=0}^{\infty} P_n = 1.$

$M/M/1$: $\rho = \lambda/\mu < 1,\ P_0 = 1-\rho\,,\ P_n = \rho^n P_0\,,\ L = \dfrac{\rho}{1-\rho}\,.$

$M/M/2$: $\lambda_n = \lambda$ for $n \ge 0,\ \mu_1 = \mu,\ \mu_n = 2\mu$ for $n \ge 2,\ \rho = \lambda/(2\mu) < 1,$

$$P_0 = \frac{1-\rho}{1+\rho}\,,\ P_n = 2\rho^n P_0 \text{ for } n \ge 1,\ L = \frac{2\rho}{1-\rho^2}\,.$$

Jackson queueing networks:
Calculate $\lambda_1, \ldots, \lambda_m$ from $\lambda_j = a_j + \sum_i \lambda_i p_{ij}$. Check that $\lambda_j < s_j\mu_j$.
Analyze each service facility (given λ_j, μ_j, s_j) to obtain $P(N_j = n_j)$.
Then $P(N_1 = n_1, \ldots, N_m = n_m) = \prod_j P(N_j = n_j)$.

$M/G/1$: $\rho = \lambda/\mu < 1,\ L_q = \dfrac{\lambda^2\sigma^2 + \rho^2}{2(1-\rho)}\,.$

A random variable ξ has an Erlang distribution with mean $1/\mu$ and shape parameter k if $\xi = \tau_1 + \cdots + \tau_k =$ a sum of k independent random variables, each with exponential distribution and mean $1/(k\mu)$.

Density function $f_\xi(t) = \dfrac{(\mu k)^k}{(k-1)!}\, t^{k-1}e^{-k\mu t}$. Standard deviation $\sigma = \dfrac{1}{\sqrt{k}\,\mu}\,.$

The deterministic EOQ model with planned shortages:

$$Q^* = \sqrt{\frac{2dK}{h}}\sqrt{\frac{p+h}{p}},\quad S^* = \sqrt{\frac{2dK}{h}}\sqrt{\frac{p}{p+h}}.$$

If shortage is not allowed then let $p \to \infty$ above.

A deterministic periodic-review model: $C_i = \min_j\{C_i^{(j)} \mid j \in \{i, \ldots, n\}\}$,

where $C_i^{(j)} = C_{j+1} + K + h(r_{i+1} + 2r_{i+2} + \cdots + (j-i)r_j)$.

A stochastic single-period model: $C(S) = cS + p\,\mathrm{E}(\xi - S)^+ + h\,\mathrm{E}(S-\xi)^+$.
If the "demand" ξ is a continuous non-negative random variable then
$\mathrm{E}(\xi-S)^+ = \int_S^{\infty}(t-S)f_\xi(t)dt$ and $\mathrm{E}(S-\xi)^+ = \int_0^S (S-t)f_\xi(t)dt.$
Moreover, $C'(S) = c + p\,(F_\xi(S)-1) + hF_\xi(S).$
If ξ is a discrete integer-valued random variable then S is also an integer and
$\mathrm{E}(\xi-S)^+ = \sum_{j=S}^{\infty}(j-S)p_\xi(j)$ and $\mathrm{E}(S-\xi)^+ = \sum_{j=0}^{S}(S-j)p_\xi(j).$
Moreover, $C(S+1) - C(S) = c + p\,(F_\xi(S)-1) + hF_\xi(S).$

The formula-sheet continues on the reverse side.

Finite horizon MDP recursion (discounting if $0 < \alpha < 1$, no discounting if $\alpha = 1$):

$$V_i^{(n)} = \min_k \{ \, C_{ik} + \alpha \textstyle\sum_j p_{ij}(k) V_j^{(n-1)} \, \} \quad \text{(backward time)}.$$

LP formulation for MDP with discounting:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_i \sum_k C_{ik} y_{ik} \\ \text{subject to} \quad & \textstyle\sum_k y_{jk} - \alpha \sum_i \sum_k p_{ij}(k) y_{ik} = \beta_j, \ \text{for all } j, \\ & y_{ik} \geq 0, \ \text{for all } i \text{ and } k. \end{aligned}$$

LP formulation for MDP without discounting:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_i \sum_k C_{ik} y_{ik} \\ \text{subject to} \quad & \textstyle\sum_i \sum_k y_{ik} = 1, \\ & \textstyle\sum_k y_{jk} - \sum_i \sum_k p_{ij}(k) y_{ik} = 0, \ \text{for all } j, \\ & y_{ik} \geq 0, \ \text{for all } i \text{ and } k. \end{aligned}$$

Policy improvement algorithm for MDP with discounting:

1. For a given policy R, defined by $(d_0, \ldots, d_M)$, calculate $V_0, \ldots, V_M$ from

$$V_i = C_{i,d_i} + \alpha \textstyle\sum_j p_{ij}(d_i) V_j, \ i = 0, \ldots, M.$$

2. R is an optimal policy if and only if

$$V_i = \min_k \{ \, C_{ik} + \alpha \textstyle\sum_j p_{ij}(k) V_j \, \}, \ \text{ for } i = 0, \ldots, M.$$

If this is not fulfilled, define a new policy R by letting, for each i,
d_i = a minimizing k above. Then go to 1.

Policy improvement algorithm for MDP without discounting:

1. For a given policy R, defined by $(d_0, \ldots, d_M)$, calculate $v_0, \ldots, v_M$ and g from

$$v_M = 0 \ \text{ and } \ g + v_i = C_{i,d_i} + \textstyle\sum_j p_{ij}(d_i) v_j, \ i = 0, \ldots, M.$$

2. R is an optimal policy if and only if

$$g + v_i = \min_k \{ \, C_{ik} + \textstyle\sum_j p_{ij}(k) v_j \, \}, \ \text{ for } i = 0, \ldots, M.$$

If this is not fulfilled, define a new policy R by letting, for each i,
d_i = a minimizing k above. Then go to 1.